# Connect RZ/A3UL Evaluation Board Kit to your Azure IoT services

# Table of Contents

# Introduction

**About this document**

This document describes how to connect RZ/A3UL Evaluation Board Kit (EVK) to Azure IoT Central and Azure IoT Hub using the Azure IoT Explorer with certified device application and device models.

IoT Plug and Play certified device simplifies the process of building devices without custom device code. Using Solution builders can integrate quickly using the certified IoT Plug and Play enabled device based on Azure IoT Central as well as third-party solutions.

This getting started guide provides step by step instruction on getting the device provisioned to Azure IoT Central and Azure IoT Hub using Device Provisioning Service (DPS) and using Azure IoT Explorer to interact with device's capabilities.

# Step 1: Prerequisites

You should have the following items ready before beginning the process:

- Prepare your development environment
  Azure IoT SDK on RZ/A3UL EVK can be used on the following environment. Please check your environment before continuing.
  Tools:
  ・IDE: e2 studio 2023-04 Windows 64-bit product version or later
  ・Tool Chain: GCC ARM A-Profile (Aarch64 bare-metal) 10.3.2021.07
  ・Flexible Software Package (FSP): RZ/A FSP Packs V2.0.1

  Refer to the following document to set up your development environment.
  2. Starting Development Introduction - Getting Started with RZ/A Flexible Software Package
  3. Set up an SMARC EVK - Getting Started with RZ/A Flexible Software Package

- Download the sample projects
  Download the sample projects from the following URL.
  RZ/A3 Azure RTOS Example Programs

- Start Serial terminal
  Start a serial terminal program (such as PuTTY, HyperTerminal or Tera Term) using the following configuration:

  > ・ Baud Rate: 115200
  > ・ Data Bits: 8
  > ・ Parity: None
  > ・ Stop Bits: 1
  > ・ Flow Control: None
  > ・ COM Port: As shown in Windows™ Device Manager.

- Setup your IoT Central
  Follow the steps below to setup your IoT Central.

  1. Create the IoT Central application.

  2. Register a new device in the IoT Central application.

  3. Save the following values generated for newly registered device:

     > ・ **ID Scope**
     > ・ **Device ID**
     > ・ **Primary Key**.

  Refer to the following document for the details.
  Quickstart - Connect a device to an Azure IoT Central application | Microsoft Learn

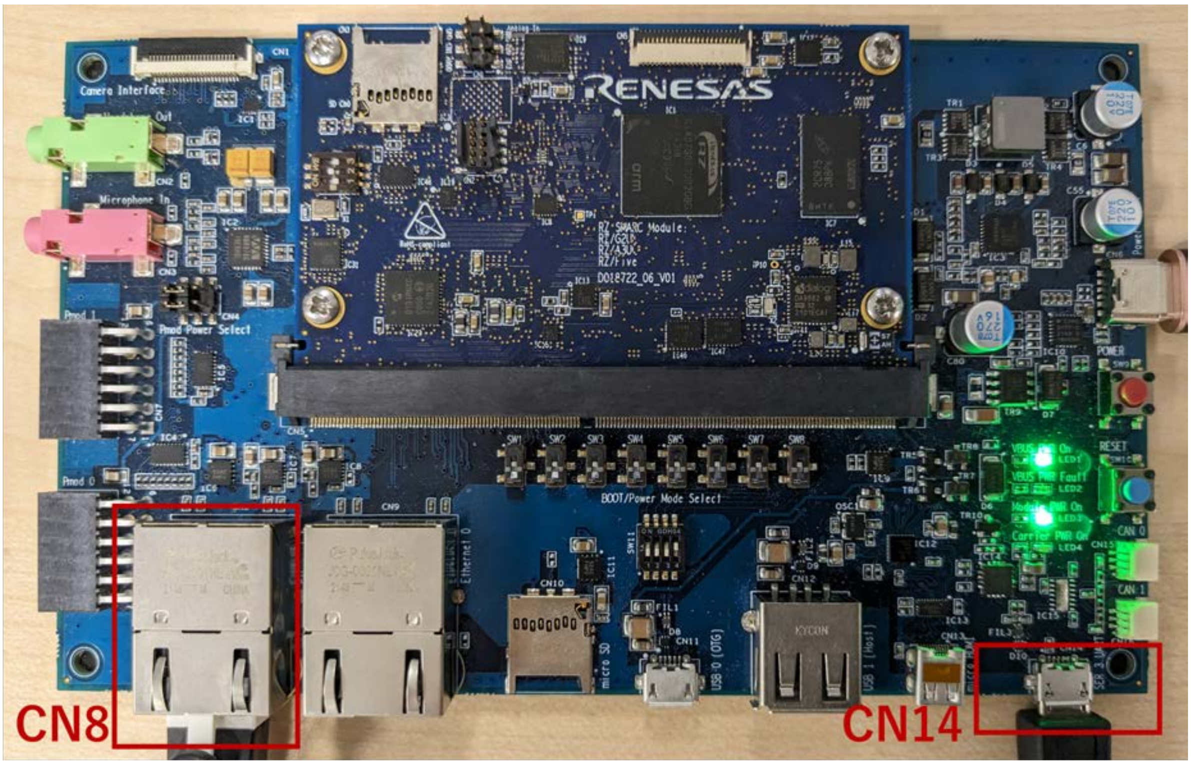
# Step 2: Prepare the Device

- Setup the device and connect power

  1. Connect the CN14 connector of the board(RZ/A3UL EVK) and the PC with a USB cable.

     > Note: By connecting to CN14 via USB, it will be recognized by the PC as UART (CDC ACM). Use the UART as a console.

  2. Connect the CN8 connector of the board and an Ethernet port with the Ethernet cable.



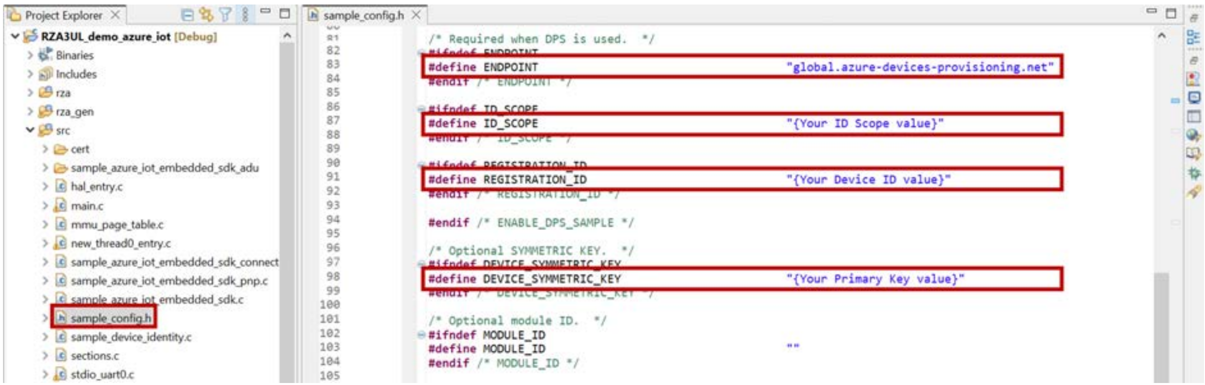
- Add configuration

    1. Extract the sample project archive and import the projects into an empty e2studio workspace. Refer to the following document for the detail.
       5.6 Importing an Existing Project into e2 studio - Getting Started with RZ/A Flexible Software Package

    2. Select the **RZA3UL_demo_azure_iot_pnp** sample and open **sample_config.h**.

    3. Set the Azure IoT device information constants to the values that you saved after you created Azure resources.

| Constant name | Value |
|---|---|
| `ENDPOINT` | "global.azure-devices-provisioning.net" |
| `ID_SCOPE` | {Your **ID Scope** value} |
| `REGISTRATION_ID` | {Your **Device ID** value} |
| `DEVICE_SYMMETRIC_KEY` | {Your **Primary Key** value} |



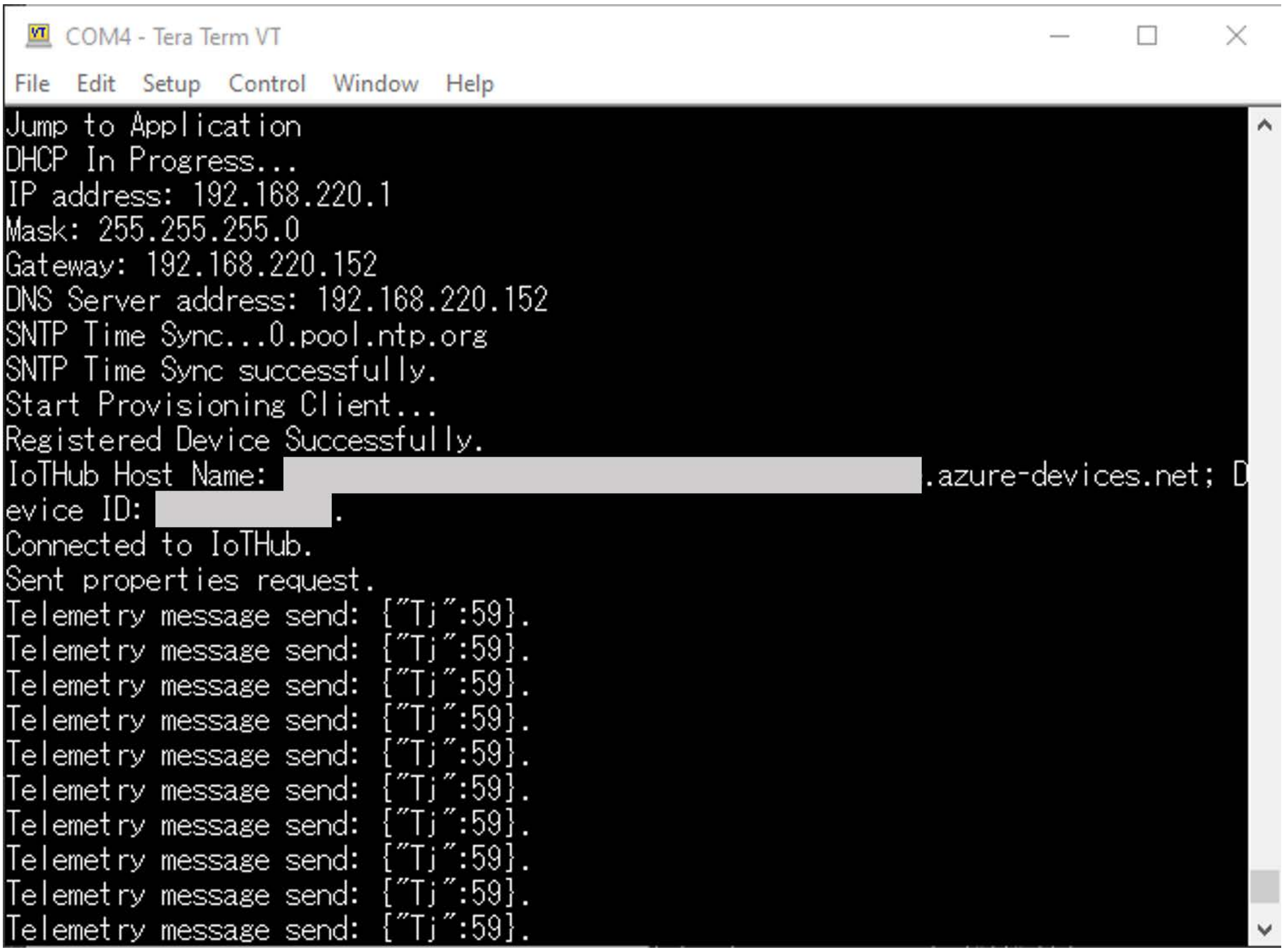
```
/* Required when DPS is used.  */
#ifndef ENDPOINT
#define ENDPOINT                                    "global.azure-devices-provisioning.net"
#endif /* ENDPOINT */

#ifndef ID_SCOPE
#define ID_SCOPE                                    "{Your ID Scope value}"
#endif /* ID_SCOPE */

#ifndef REGISTRATION_ID
#define REGISTRATION_ID                             "{Your Device ID value}"
#endif /* REGISTRATION_ID */

#endif /* ENABLE_DPS_SAMPLE */

/* Optional SYMMETRIC KEY.  */
#ifndef DEVICE_SYMMETRIC_KEY
#define DEVICE_SYMMETRIC_KEY                        "{Your Primary Key value}"
#endif /* DEVICE_SYMMETRIC_KEY */

/* Optional module ID.  */
#ifndef MODULE_ID
#define MODULE_ID                                   ""
#endif /* MODULE_ID */
```

# Step 3: Build SDK and Run Samples

Refer to following document to build and debug the project "RZA3UL_demo_azure_iot".
4.4 Build the Blinky Project - Getting Started with RZ/A Flexible Software Package
5.4 Debugging the Project - Getting Started with RZ/A Flexible Software Package

> Note: Due to the bug of FSP, some project contents must be modified to execute this sample project. They have been already modified for the released project, but don't delete "rza" folder and regenerate it.

After running the projects, confirm that following message is displayed on the terminal software.



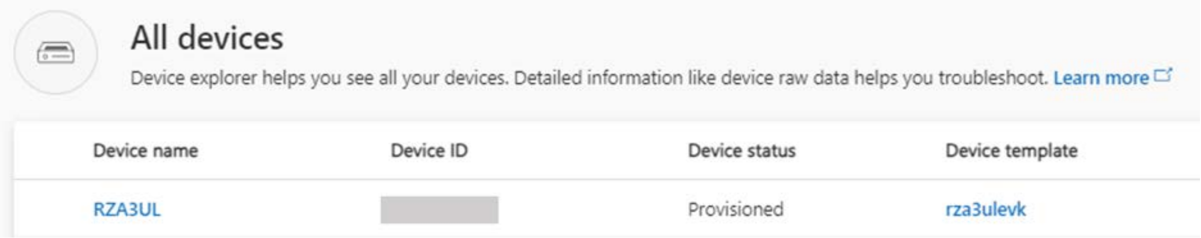
```
Jump to Application
DHCP In Progress...
IP address: 192.168.220.1
Mask: 255.255.255.0
Gateway: 192.168.220.152
DNS Server address: 192.168.220.152
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync successfully.
Start Provisioning Client...
Registered Device Successfully.
IoTHub Host Name: [redacted].azure-devices.net; D
evice ID: [redacted].
Connected to IoTHub.
Sent properties request.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
Telemetry message send: {"Tj":59}.
```

# Step 4: Use Azure IoT Central

IoT Central is an IoT application platform that reduces the cost and complexity of creating and managing IoT solutions.

- Verify the device status
  To view the device status in IoT Central portal:

  1. From the application dashboard, select **Devices** on the side navigation menu.
  2. Check the **Device status** of the device is updated to **Provisioned**.
  3. Check the **Device template** of the device has updated to **rza3ulevk**.



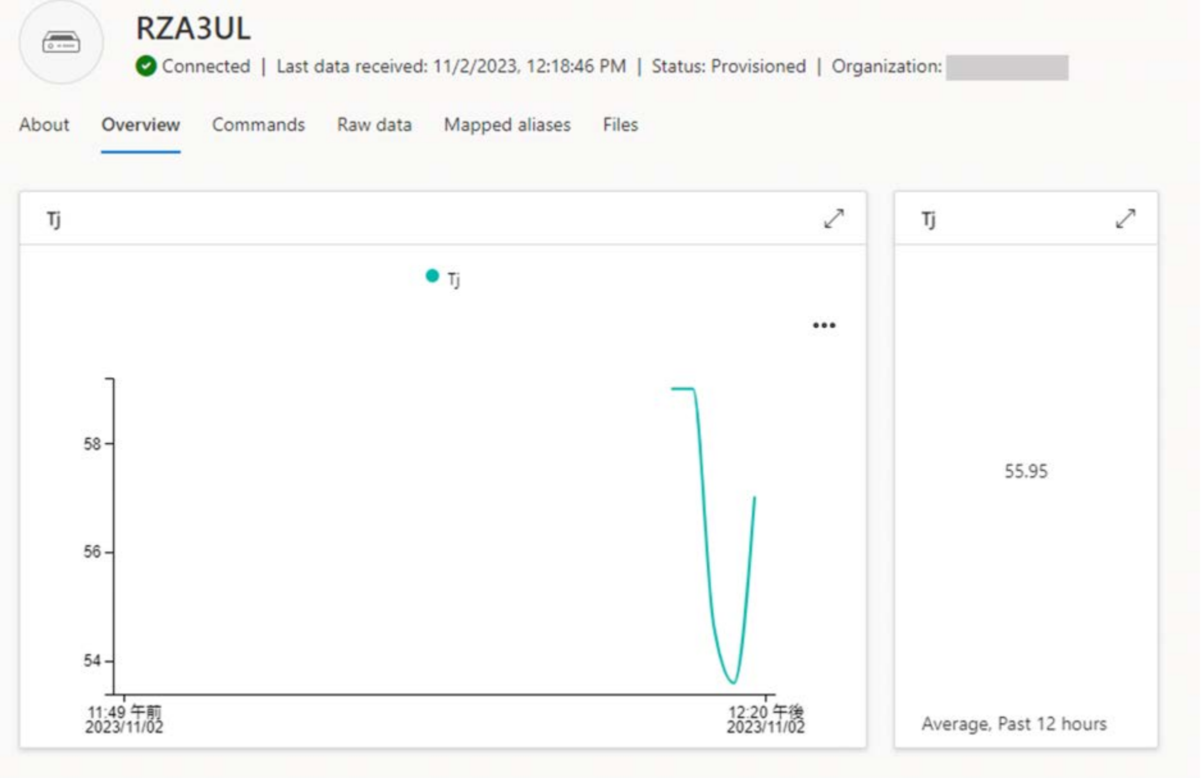
All devices

Device explorer helps you see all your devices. Detailed information like device raw data helps you troubleshoot. Learn more

| Device name | Device ID | Device status | Device template |
|---|---|---|---|
| RZA3UL | [redacted] | Provisioned | rza3ulevk |

- View device telemetry
  With IoT Central, you can view the flow of telemetry from your device to the cloud. In this sample project, Tj (temperature inside the LSI measured by thermal sensor unit (TSU)) is sent as telemetry.
  To view telemetry in IoT Central portal:

  1. From the application dashboard, select **Devices** on the side navigation menu.
  2. Select the device from the device list.
  3. View the telemetry as the device sends messages to the cloud in the **Overview** tab.

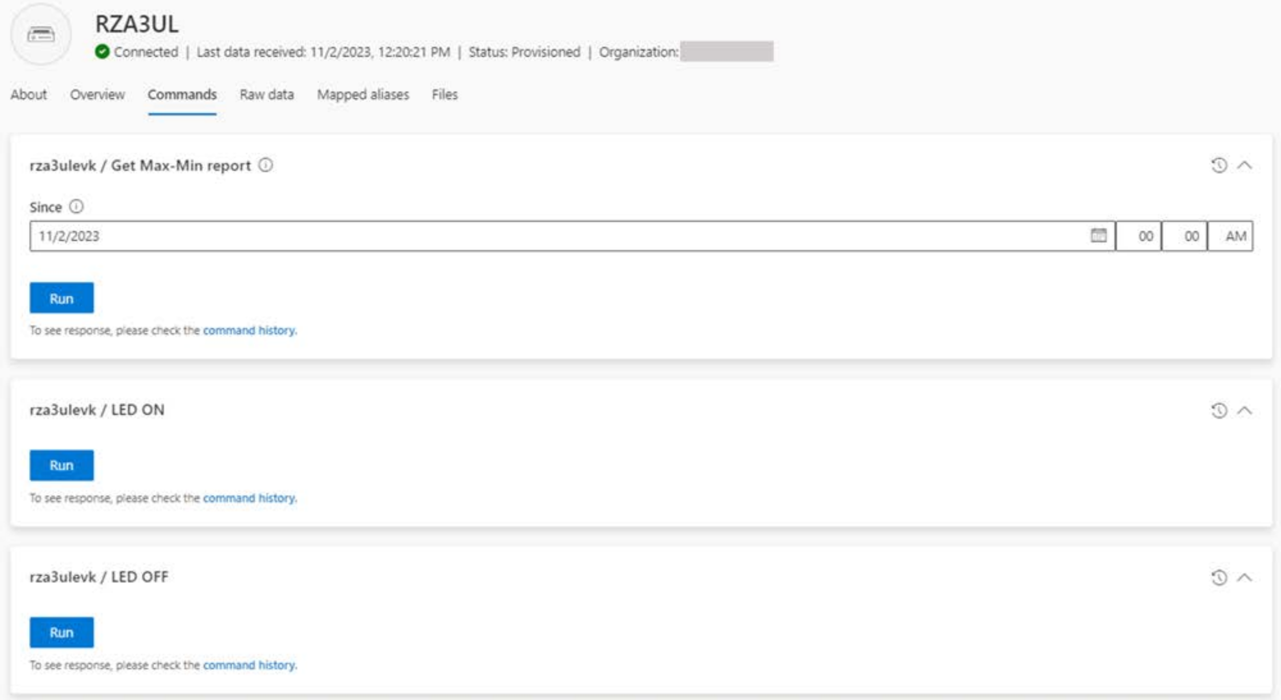


RZA3UL

Connected | Last data received: 11/2/2023, 12:18:46 PM | Status: Provisioned | Organization: [redacted]

- Send commands
  On the **Command** tab, you can send IoT Plug and Play command **getMaxMinReport**, **LED_ON** and **LED_OFF** to the device.

  > Note: If you check the function of command **LED_ON** or **LED_OFF**, connect Pmod(LED) to J1 connector(7:12) on the board.



RZA3UL

Connected | Last data received: 11/2/2023, 12:20:21 PM | Status: Provisioned | Organization: [redacted]

rza3ulevk / Get Max-Min report

Since: 11/2/2023 00 00 AM

To see response, please check the command history.

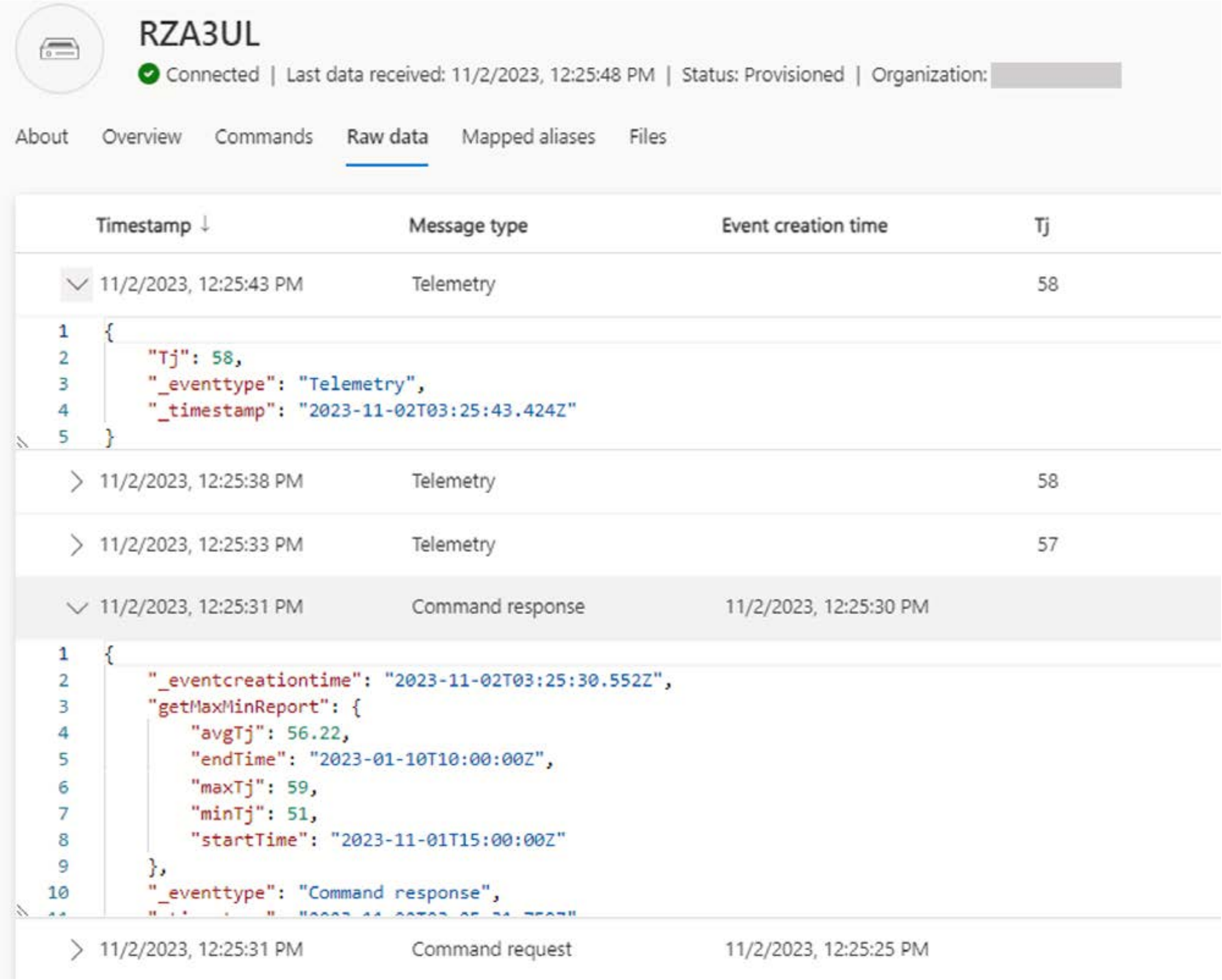
rza3ulevk / LED ON

To see response, please check the command history.

rza3ulevk / LED OFF

To see response, please check the command history.

- View raw data
  On the **Raw data** tab, you can view the raw data such as **Telemetry** and **Command response**.



RZA3UL

Connected | Last data received: 11/2/2023, 12:25:48 PM | Status: Provisioned | Organization: [redacted]

| Timestamp ↓ | Message type | Event creation time | Tj |
|---|---|---|---|
| 11/2/2023, 12:25:43 PM | Telemetry | | 58 |

```
{
    "Tj": 58,
    "_eventtype": "Telemetry",
    "_timestamp": "2023-11-02T03:25:43.424Z"
}
```

| Timestamp ↓ | Message type | Event creation time | Tj |
|---|---|---|---|
| 11/2/2023, 12:25:38 PM | Telemetry | | 58 |
| 11/2/2023, 12:25:33 PM | Telemetry | | 57 |
| 11/2/2023, 12:25:31 PM | Command response | 11/2/2023, 12:25:30 PM | |

```
{
    "_eventcreationtime": "2023-11-02T03:25:30.552Z",
    "getMaxMinReport": {
        "avgTj": 56.22,
        "endTime": "2023-01-10T10:00:00Z",
        "maxTj": 59,
        "minTj": 51,
        "startTime": "2023-11-01T15:00:00Z"
    },
    "_eventtype": "Command response",
```

| Timestamp ↓ | Message type | Event creation time | Tj |
|---|---|---|---|
| 11/2/2023, 12:25:31 PM | Command request | 11/2/2023, 12:25:25 PM | |

# Step 5: Integration with Azure IoT Explorer

> Note: This section is advanced.

You can use the Azure IoT Explorer to view and manage the properties of your devices. In the following steps, you'll add a connection to your IoT hub in IoT Explorer. With the connection, you can view properties for devices associated with the IoT hub.

Download and install latest Azure IoT Explorer from the following URL.
https://github.com/Azure/azure-iot-explorer/releases

- Setup your IoT hub
  Follow the steps below to setup your IoT hub.

  1. Create an IoT hub.

  2. Register a new device in the IoT hub.

  Refer to the following document for the details.
  Use the Azure portal to create an IoT Hub | Microsoft Learn

- Setup a DPS
  The IoT Hub Device Provisioning Service (DPS) is a helper service for IoT Hub that enables zero-touch, just-in-time provisioning devices to the right IoT hub in a secure and scalable manner. In the following steps, you will enroll the board in DPS using Symmetric Key and provision it automatically in IoT Hub when connecting to the Internet.

  Follow the steps below to setup a DPS.

  1. Create a DPS to provision the device in IoT Hub automatically.

  2. Link an IoT hub to the DPS instance.

  3. Add an individual enrollment record in DPS. Your device can use it to connect to DPS and perform the provisioning in IoT Hub.

  4. Save the following values:

     > - **ID Scope** of your Device Provisioning Service
     > - **Registration ID** for your device enrollment
     > - **Primary Symmetric Key** for your device enrollment.

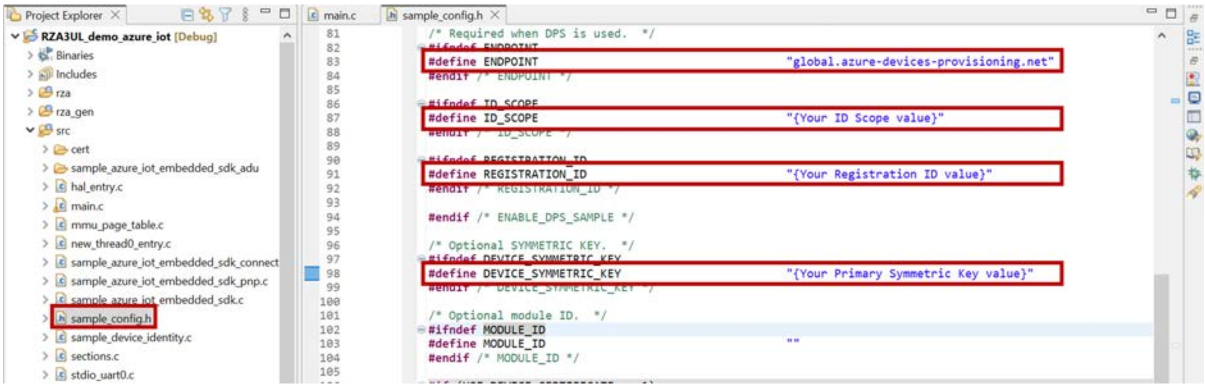
  Refer to the following documents for the details.
  Quickstart - Set up Device Provisioning Service in portal | Microsoft Learn
  Manage device enrollments for Azure IoT Hub Device Provisioning Service in the Azure portal | Microsoft Learn

- Add configuration

    1. Select the **RZA3UL_demo_azure_iot_pnp** sample in the e2studio workspace and open **sample_config.h**.

    2. Set the Azure IoT device information constants to the values that you saved after you created Azure resources.

| Constant name | Value |
|---|---|
| `ENDPOINT` | "global.azure-devices-provisioning.net" |
| `ID_SCOPE` | {Your **ID Scope** value} |
| `REGISTRATION_ID` | {Your **Registration ID** value} |
| `DEVICE_SYMMETRIC_KEY` | {Your **Primary Symmetric Key** value} |



```
/* Required when DPS is used.  */
#ifndef ENDPOINT
#define ENDPOINT                                  "global.azure-devices-provisioning.net"
#endif /* ENDPOINT */

#ifndef ID_SCOPE
#define ID_SCOPE                                  "{Your ID Scope value}"
#endif /* ID_SCOPE */

#ifndef REGISTRATION_ID
#define REGISTRATION_ID                           "{Your Registration ID value}"
#endif /* REGISTRATION_ID */

#endif /* ENABLE_DPS_SAMPLE */

/* Optional SYMMETRIC KEY.  */
#ifndef DEVICE_SYMMETRIC_KEY
#define DEVICE_SYMMETRIC_KEY                      "{Your Primary Symmetric Key value}"
#endif /* DEVICE_SYMMETRIC_KEY */

/* Optional module ID.  */
#ifndef MODULE_ID
#define MODULE_ID                                 ""
#endif /* MODULE_ID */
```

- Build and run the sample
  Follow Step 3: Build SDK and Run Samples to build and run the sample project.

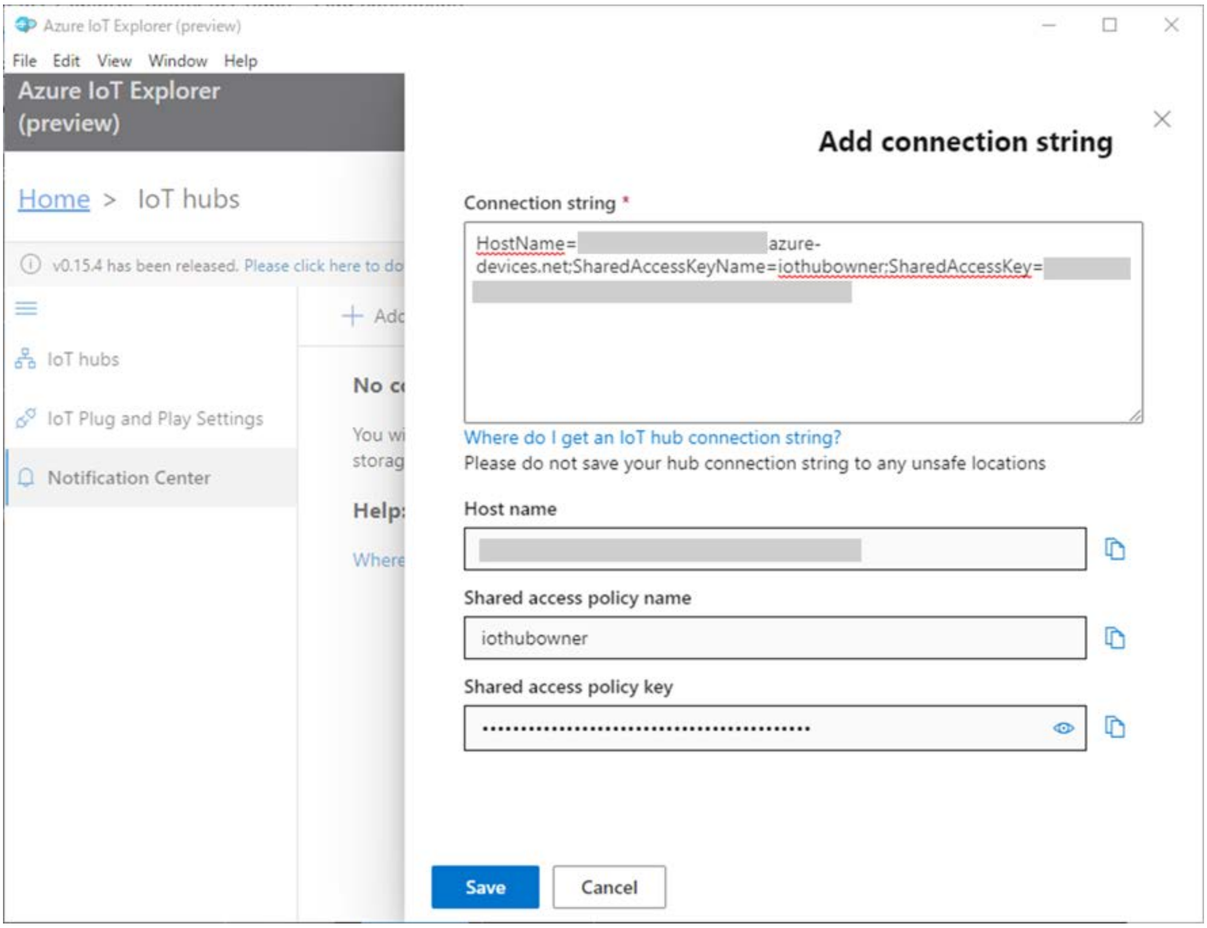
- Setup the Azure IoT Explorer
  To add a connection to your IoT hub:

    1. Get the connection string for your IoT hub.

    2. In Azure IoT Explorer, select **IoT hubs** > **Add connection**.

    3. Paste the connection string into the **Connection string** box.

4. Select Save.


Azure IoT Explorer (preview)
File Edit View Window Help
Azure IoT Explorer (preview)
Home > IoT hubs
v0.15.4 has been released. Please click here to do
IoT hubs
IoT Plug and Play Settings
Notification Center
+ Add
No c
You wi
storag
Help:
Where
Save
Cancel

Add connection string

Connection string *

HostName=[redacted]azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=[redacted]

Where do I get an IoT hub connection string?

Please do not save your hub connection string to any unsafe locations

Host name

[redacted]

Shared access policy name

iothubowner

Shared access policy key

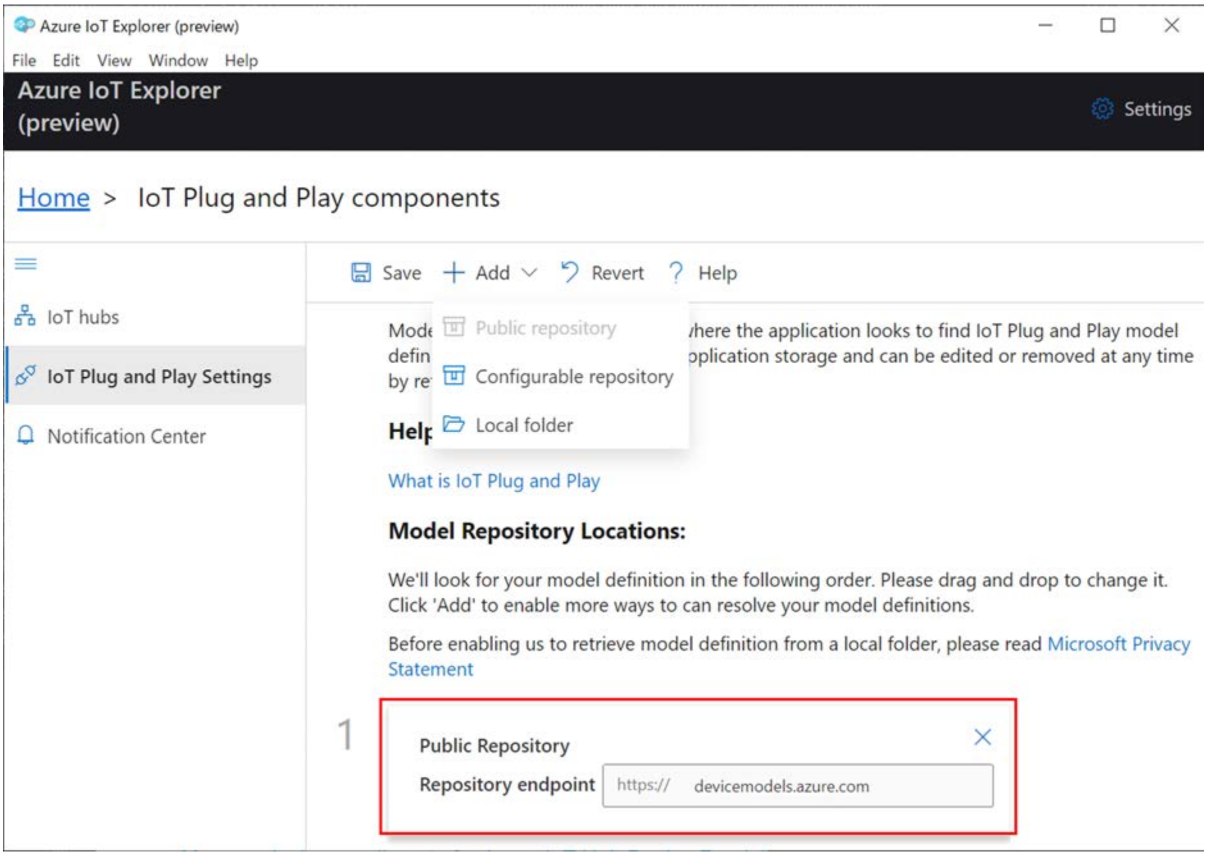
••••••••••••••••••••••••••••••••••••••••••••

If the connection succeeds, the Azure IoT Explorer switches to a Devices view and lists your device.

To set IoT Plug and Play Settings:

5. In Azure IoT Explorer, select **IoT hubs** > **IoT Plug and Play Settings**.

6. Add **Public repository** to use IoT Plug and Play Settings.


Azure IoT Explorer (preview)
File Edit View Window Help
Azure IoT Explorer (preview)
Settings
Home > IoT Plug and Play components
IoT hubs
IoT Plug and Play Settings
Notification Center
Save
Add
Revert
Help
Public repository
Configurable repository
Local folder
What is IoT Plug and Play
1

Mode… defin… by re… …here the application looks to find IoT Plug and Play model …pplication storage and can be edited or removed at any time

Help…

**Model Repository Locations:**

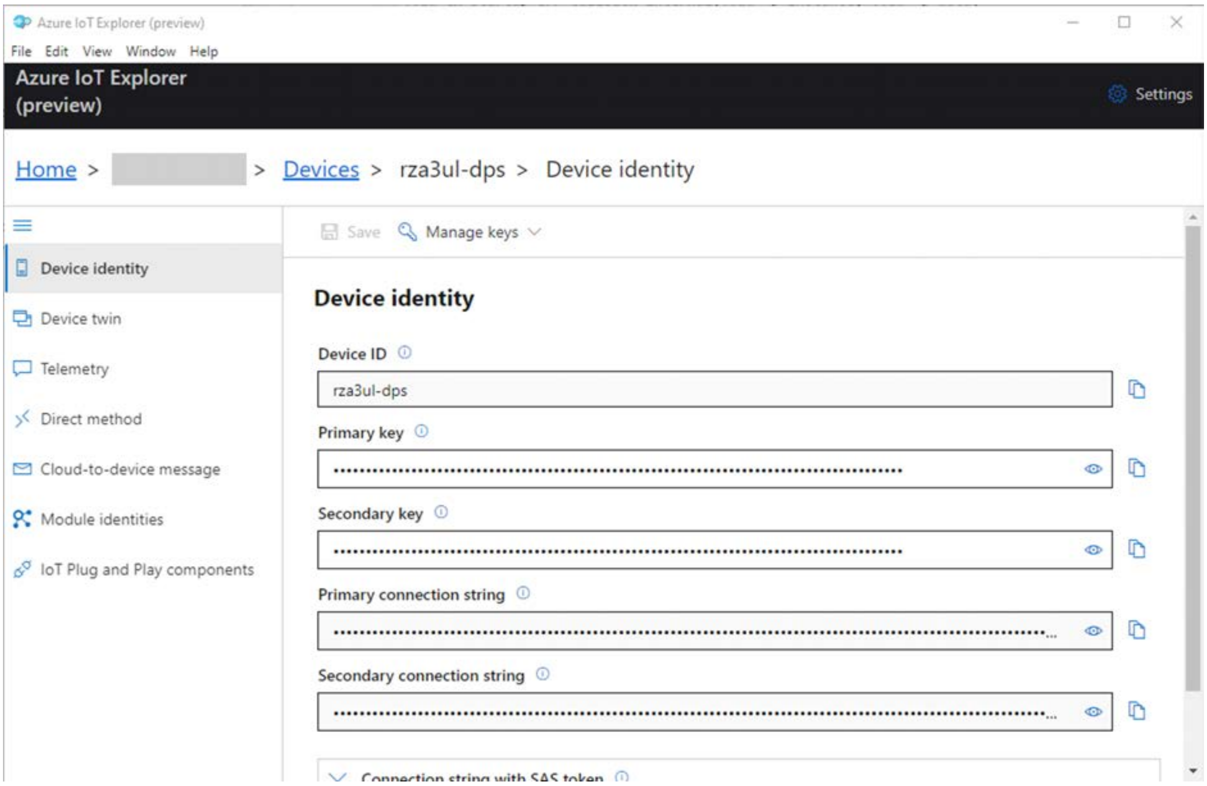
We'll look for your model definition in the following order. Please drag and drop to change it. Click 'Add' to enable more ways to can resolve your model definitions.

Before enabling us to retrieve model definition from a local folder, please read Microsoft Privacy Statement

Public Repository
Repository endpoint: https:// devicemodels.azure.com

To view device properties using Azure IoT Explorer:

1. Select the link for your device identity. IoT Explorer displays details for the devices.


Azure IoT Explorer (preview)
File Edit View Window Help
Azure IoT Explorer (preview)
Settings
Home > [redacted] > Devices > rza3ul-dps > Device identity
Save
Manage keys
Device identity
Device twin
Telemetry
Direct method
Cloud-to-device message
Module identities
IoT Plug and Play components

Device identity

Device ID: rza3ul-dps

Primary key: ••••••••••••••••••••••••••••••••••••••••••••

Secondary key: ••••••••••••••••••••••••••••••••••••••••••••

Primary connection string: ••••••••••••••••••••••••••••••••••••••••••••...

Secondary connection string: ••••••••••••••••••••••••••••••••••••••••••••...

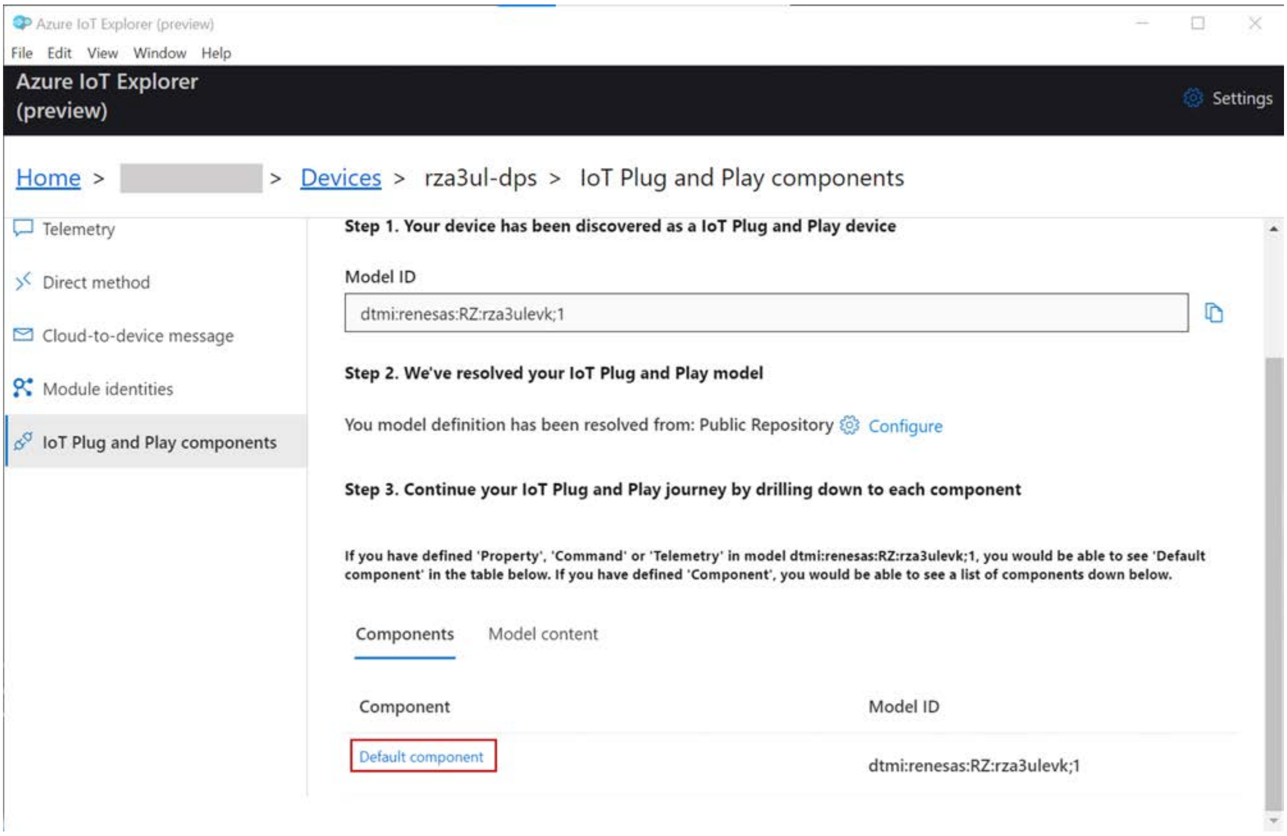
Connection string with SAS token

2. Inspect the properties for your device in the **Device identity** panel.

# Step 6: Interact with IoT Plug and Play components

> Note: This section is advanced.

1. Select **IoT Plug and Play components** and **Default component**.


Azure IoT Explorer (preview)
File Edit View Window Help
Azure IoT Explorer (preview)
Settings
Home > [redacted] > Devices > rza3ul-dps > IoT Plug and Play components
Telemetry
Direct method
Cloud-to-device message
Module identities
IoT Plug and Play components
Configure
Components
Model content

**Step 1. Your device has been discovered as a IoT Plug and Play device**

Model ID: dtmi:renesas:RZ:rza3ulevk;1

**Step 2. We've resolved your IoT Plug and Play model**

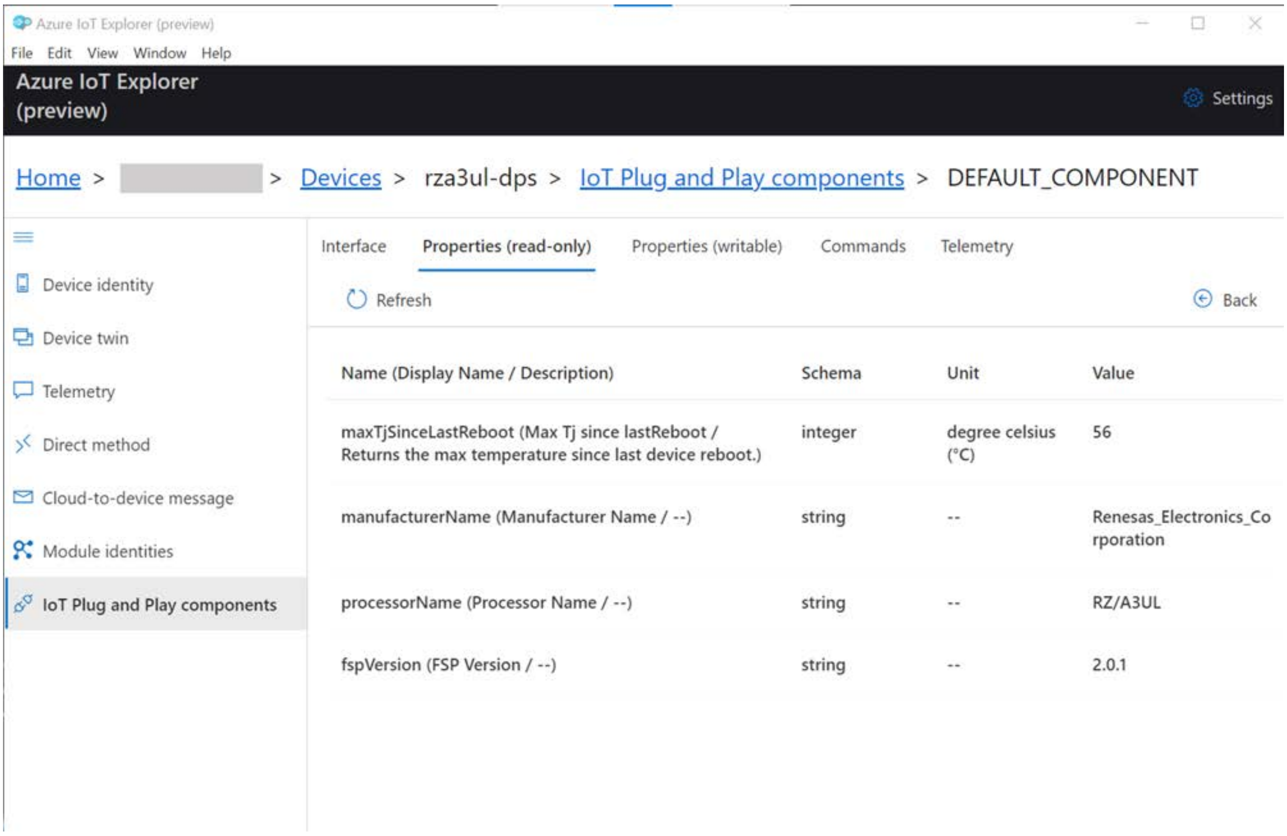
You model definition has been resolved from: Public Repository

**Step 3. Continue your IoT Plug and Play journey by drilling down to each component**

If you have defined 'Property', 'Command' or 'Telemetry' in model dtmi:renesas:RZ:rza3ulevk;1, you would be able to see 'Default component' in the table below. If you have defined 'Component', you would be able to see a list of components down below.

| Component | Model ID |
|---|---|
| Default component | dtmi:renesas:RZ:rza3ulevk;1 |

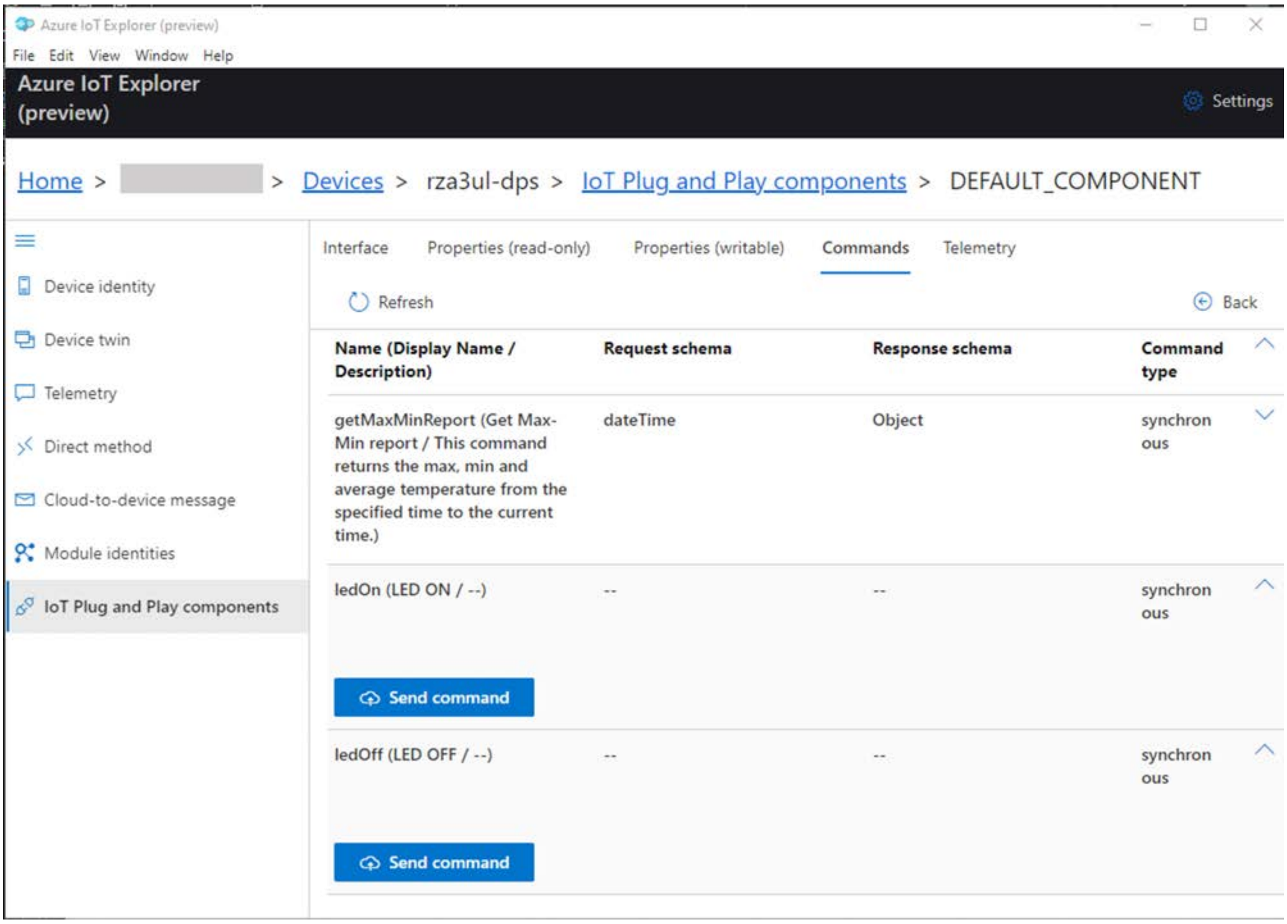
2. Select **Properties(read-only)**. You can see the device properties.


Azure IoT Explorer (preview)
File Edit View Window Help
Azure IoT Explorer (preview)
Settings
Home > [redacted] > Devices > rza3ul-dps > IoT Plug and Play components > DEFAULT_COMPONENT
Device identity
Device twin
Telemetry
Direct method
Cloud-to-device message
Module identities
IoT Plug and Play components
Interface
Properties (read-only)
Properties (writable)
Commands
Telemetry
Refresh
Back

| Name (Display Name / Description) | Schema | Unit | Value |
|---|---|---|---|
| maxTjSinceLastReboot (Max Tj since lastReboot / Returns the max temperature since last device reboot.) | integer | degree celsius (°C) | 56 |
| manufacturerName (Manufacturer Name / --) | string | -- | Renesas_Electronics_Corporation |
| processorName (Processor Name / --) | string | -- | RZ/A3UL |
| fspVersion (FSP Version / --) | string | -- | 2.0.1 |

3. Select **Commands**. You can send IoT Plug and Play command **getMaxMinReport**, **LED_ON** and **LED_OFF** to the device.

> Note: If you check the function of command **LED_ON** or **LED_OFF**, connect Pmod(LED) to J1 connector(7:12) on the board.


Azure IoT Explorer (preview)
File Edit View Window Help
Azure IoT Explorer (preview)
Settings
Home > [redacted] > Devices > rza3ul-dps > IoT Plug and Play components > DEFAULT_COMPONENT
Device identity
Device twin
Telemetry
Direct method
Cloud-to-device message
Module identities
IoT Plug and Play components
Interface
Properties (read-only)
Properties (writable)
Commands
Telemetry
Refresh
Back
Send command
Send command

| Name (Display Name / Description) | Request schema | Response schema | Command type |
|---|---|---|---|
| getMaxMinReport (Get Max-Min report / This command returns the max, min and average temperature from the specified time to the current time.) | dateTime | Object | synchronous |
| ledOn (LED ON / --) | -- | -- | synchronous |
| ledOff (LED OFF / --) | -- | -- | synchronous |